subspace clustering

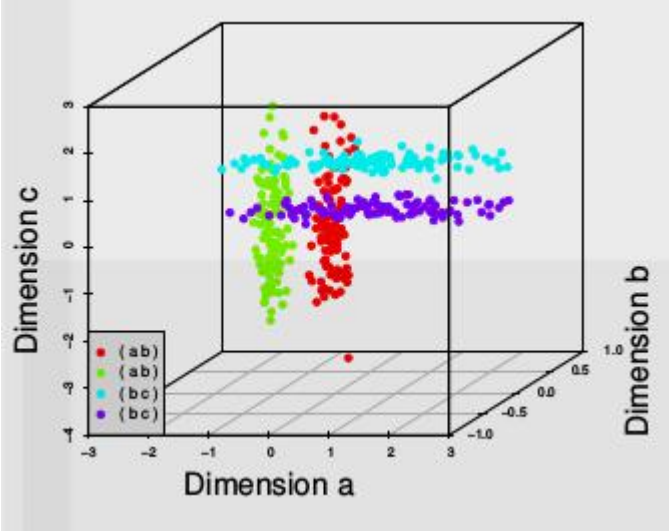
xuwf

2017.11.17

Intuition

- Definition

Subspace clustering is an extension of feature selection that attempts to find clusters in different subspaces of the same dataset.



Data：The dataset is divided into four clusters of 100 instances, each existing in only two of the three dimensions.

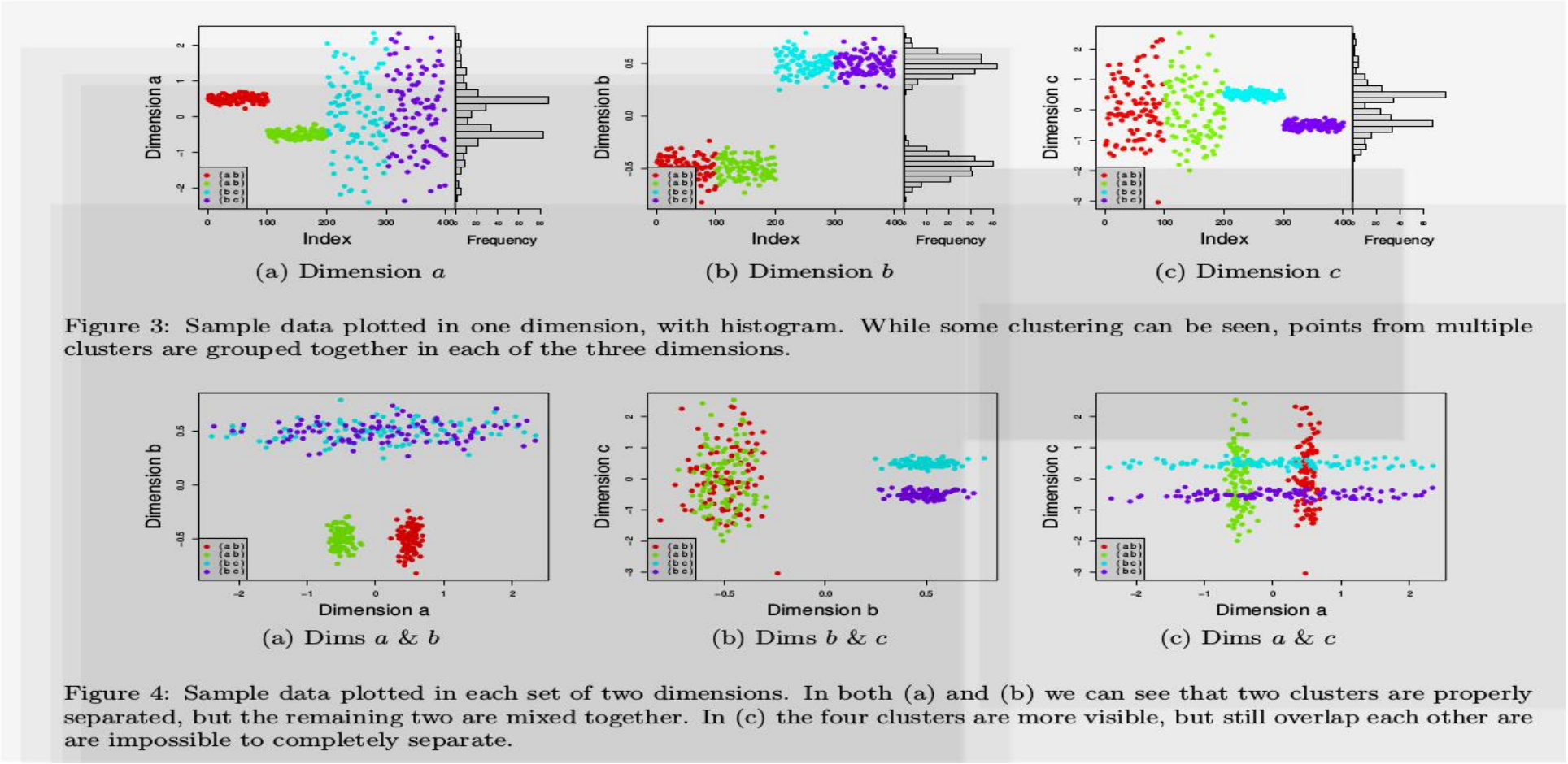
Difficulty:In higher dimensional datasets this problem becomes even worse and the clusters become impossible to find, suggesting that we consider fewer dimensions.

Note:PCA is not work!since relative distances are preserved and the effects of the irrelevant dimension remain.

Intuition

Thus, the key to finding each of the clusters in this dataset is to look in the appropriate subspaces.

How to: we might try using a feature selection algorithm to remove one or two dimensions.



(a) Dimension a (b) Dimension b (c) Dimension c

Figure 3: Sample data plotted in one dimension, with histogram. While some clustering can be seen, points from multiple clusters are grouped together in each of the three dimensions.

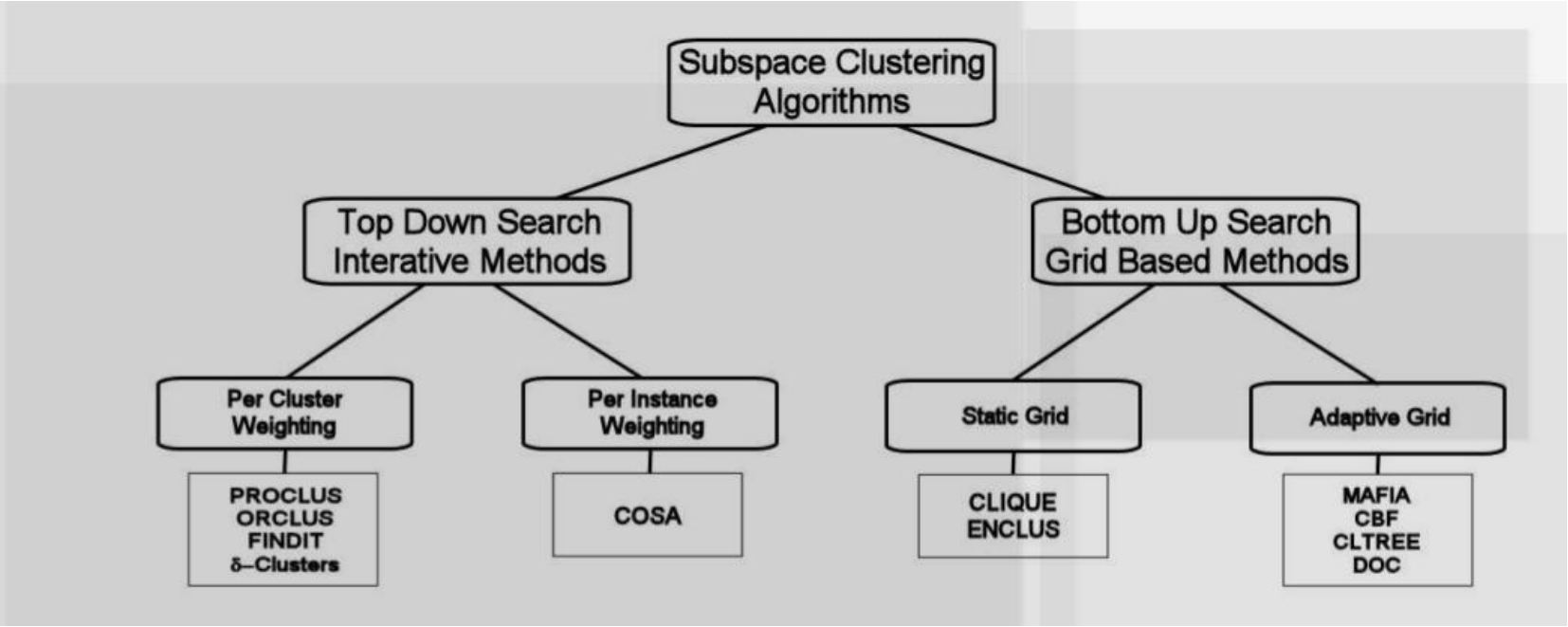
(a) Dims a & b (b) Dims b & c (c) Dims a & c

Figure 4: Sample data plotted in each set of two dimensions. In both (a) and (b) we can see that two clusters are properly separated, but the remaining two are mixed together. In (c) the four clusters are more visible, but still overlap each other are are impossible to completely separate.

Methods

Naive approach: A naive approach might be to search through all possible subspaces and use cluster validation techniques to determine the subspaces with the best clusters

- **sophisticated heuristic search**



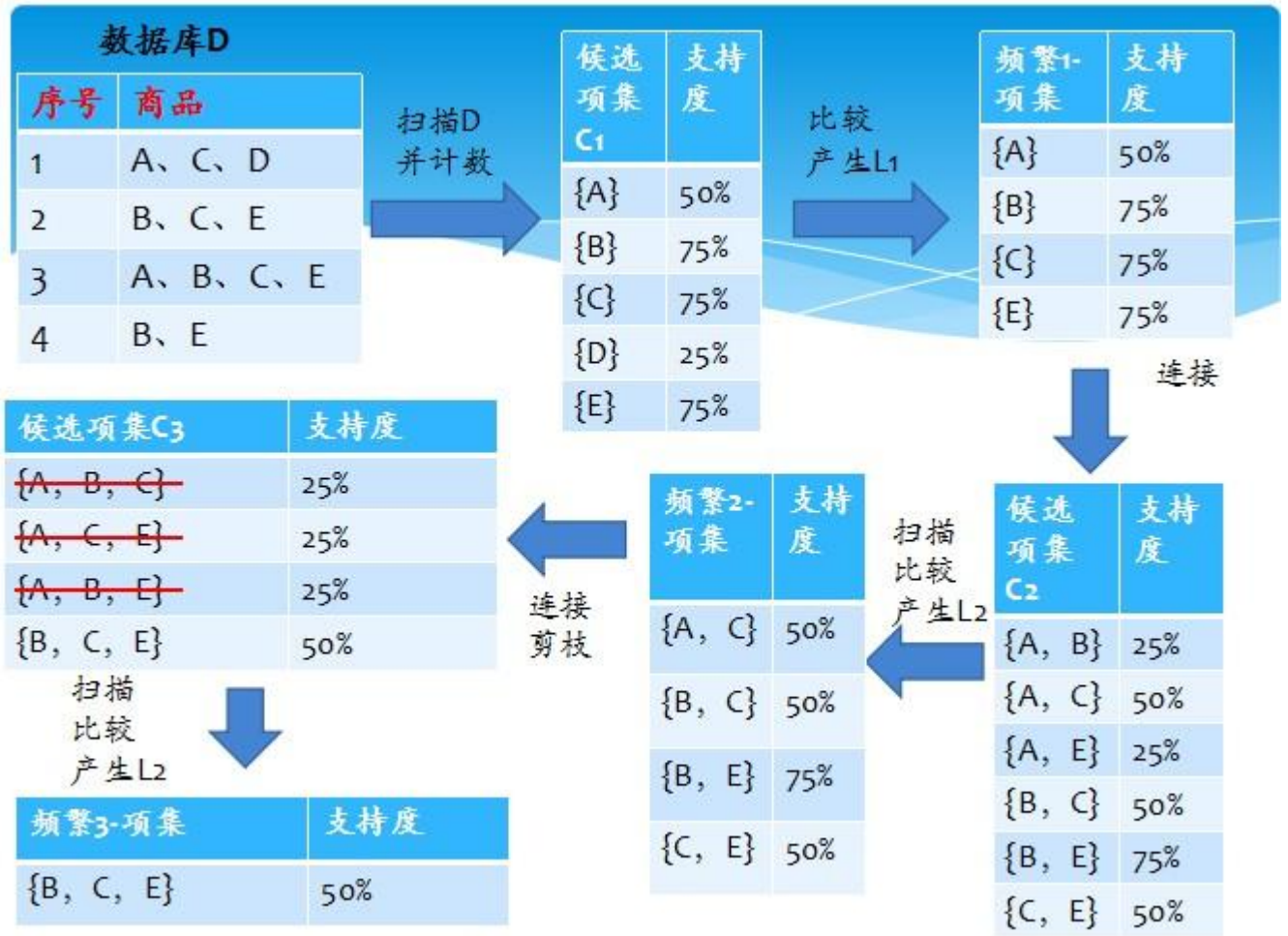
Bottom-Up Subspace Search Methods

an APRIORI style approach

Algorithms first create a histogram for each dimension and selecting those bins with densities above a given threshold.

MAFIA:based on grid

two parameter: Step size of the grid,threshold of density



MAFIA

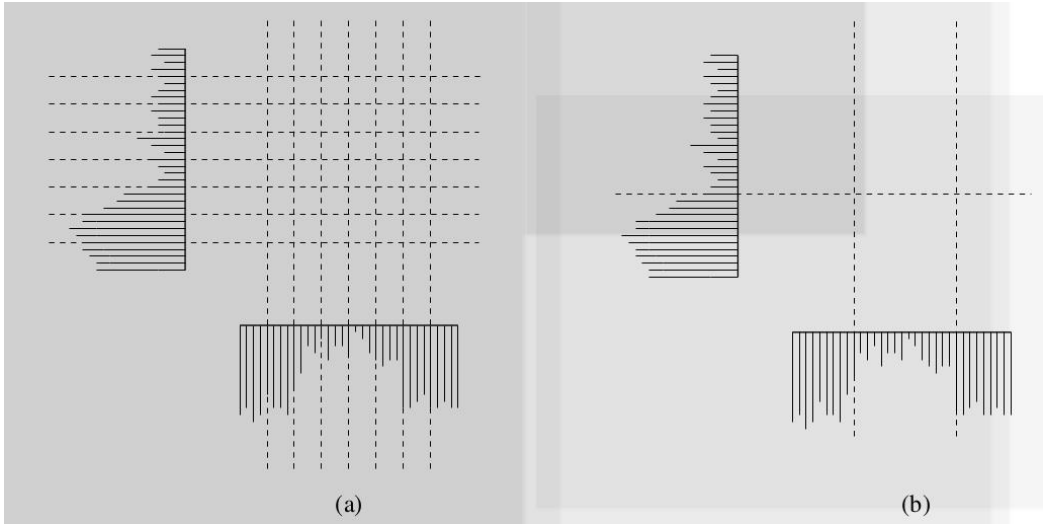


Figure 1: (a) Uniform grid size (b) Adaptive grid size

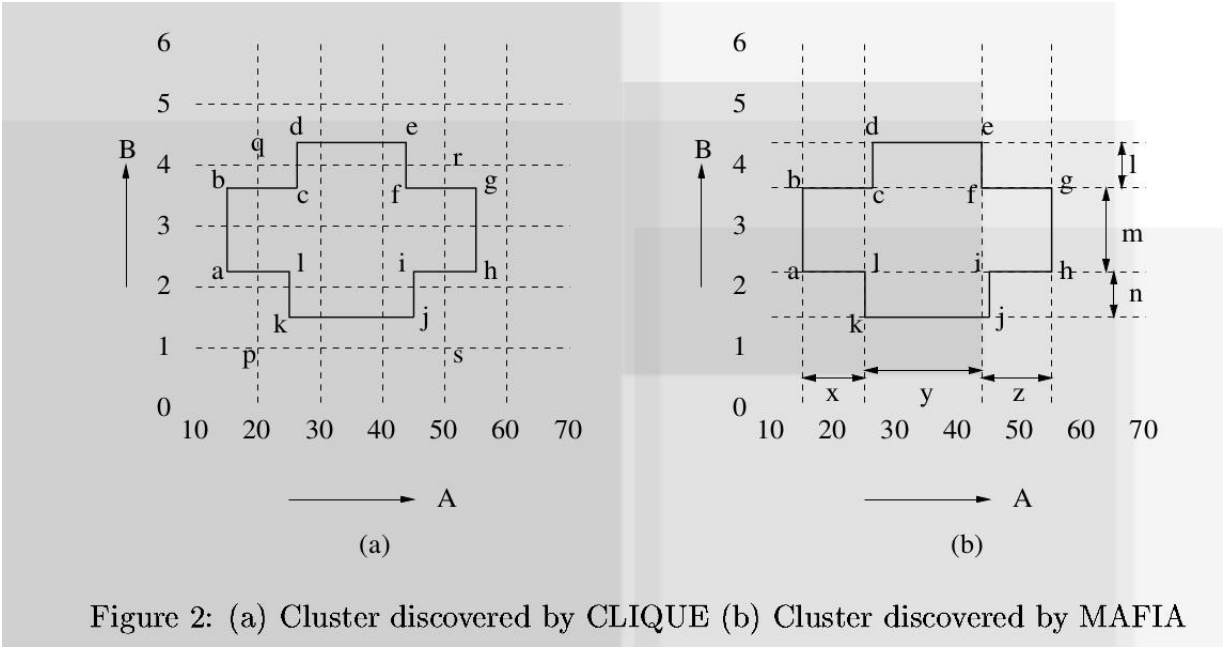


Figure 2: (a) Cluster discovered by CLIQUE (b) Cluster discovered by MAFIA

sparse subspace clustering

- Spectral clustering-based methods.

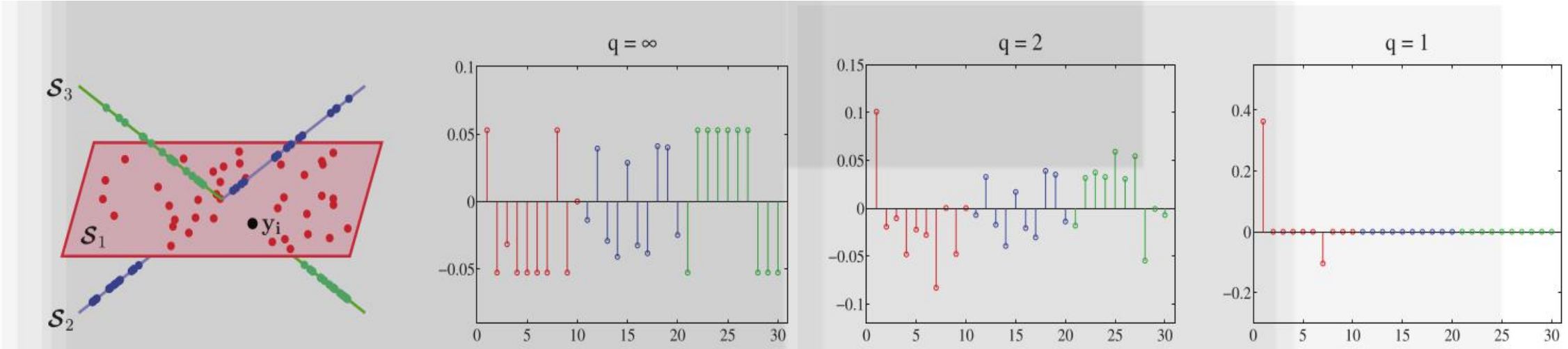


Fig. 3. Three subspaces in $\mathbb{R}^3$ with 10 data points in each subspace, ordered such that the fist and the last 10 points belong to $\mathcal{S}_1$ and $\mathcal{S}_3$, respectively. The solution of the ℓ_q-minimization program in (3) for $\boldsymbol{y}_i$ lying in $\mathcal{S}_1$ for $q = 1, 2, \infty$ is shown. Note that as the value of q decreases, the sparsity of the solution increases. For $q = 1$, the solution corresponds to choosing two other points lying in $\mathcal{S}_1$.

sparse subspace clustering

- symbols

Let $\{\mathcal{S}_\ell\}_{\ell=1}^n$ be an arrangement of n linear subspaces of $\mathbb{R}^D$ of dimensions $\{d_\ell\}_{\ell=1}^n$. Consider a given collection of N noise-free data points $\{\boldsymbol{y}_i\}_{i=1}^N$ that lie in the union of the n subspaces. Denote the matrix containing all the data points as

$$\boldsymbol{Y} \triangleq [\boldsymbol{y}_1 \ \dots \ \boldsymbol{y}_N] = [\boldsymbol{Y}_1 \ \dots \ \boldsymbol{Y}_n]\boldsymbol{\Gamma}, \tag{1}$$

where $\boldsymbol{Y}_\ell \in \mathbb{R}^{D \times N_\ell}$ is a rank-d_ℓ matrix of the $N_\ell > d_\ell$ points that lie in $\mathcal{S}_\ell$ and $\boldsymbol{\Gamma} \in \mathbb{R}^{N \times N}$ is an unknown permutation matrix. We assume that we do not know a priori the bases of the subspaces nor do we know which data points belong to which subspace. The *subspace clustering* problem refers to the problem of finding the number of subspaces, their dimensions, a basis for each subspace, and the segmentation of the data from $\boldsymbol{Y}$.

sparse subspace clustering

- self-expressiveness

each data point in a union of subspaces can be efficiently reconstructed by a combination of other points in the dataset.

More precisely, each data point $\boldsymbol{y}_i \in \cup_{\ell=1}^{n} \mathcal{S}_\ell$ can be written as

$$\boldsymbol{y}_i = \boldsymbol{Y}\boldsymbol{c}_i, \quad c_{ii} = 0, \tag{2}$$

where $\boldsymbol{c}_i \triangleq [c_{i1}\ c_{i2}\ \ldots\ c_{iN}]^\top$ and the constraint $c_{ii} = 0$ eliminates the trivial solution of writing a point as a linear combination of itself. In other words, the matrix of data points $\boldsymbol{Y}$ is a self-expressive dictionary in which each point can be written as a linear combination of other points.

sparse subspace clustering

there exists a sparse solution, $\boldsymbol{c}_i$, whose nonzero entries correspond to data points from the same subspace as $\boldsymbol{y}_i$. We refer to such a solution as a subspace-sparse representation.

More specifically, a data point $\boldsymbol{y}_i$ that lies in the d_ℓ-dimensional subspace $\mathcal{S}_\ell$ can be written as a linear combination of d_ℓ other points in general directions from $\mathcal{S}_\ell$. As a result, ideally, a sparse representation of a data point finds points from the same subspace where the number of the nonzero elements corresponds to the dimension of the underlying subspace.

For a system of equations such as (2) with infinitely many solutions, one can restrict the set of solutions by minimizing an objective function such as the ℓ_q-norm of the solution[1] as

$$\min \|\boldsymbol{c}_i\|_q \quad \text{s.t.} \quad \boldsymbol{y}_i = \boldsymbol{Y}\boldsymbol{c}_i, \; c_{ii} = 0. \tag{3}$$

sparse subspace clustering

$$\min \|\boldsymbol{c}_i\|_1 \quad \text{s.t.} \quad \boldsymbol{y}_i = \boldsymbol{Y}\boldsymbol{c}_i,\ c_{ii} = 0, \tag{4}$$

which can be solved efficiently using convex programming tools [48], [49], [50] and is known to prefer sparse solutions [29], [30], [31].

We can also rewrite the sparse optimization program (4) for all data points $i = 1, \ldots, N$ in matrix form as

$$\min \|\boldsymbol{C}\|_1 \quad \text{s.t.} \quad \boldsymbol{Y} = \boldsymbol{Y}\boldsymbol{C}, \operatorname{diag}(\boldsymbol{C}) = \boldsymbol{0}, \tag{5}$$

where $\boldsymbol{C} \triangleq [\boldsymbol{c}_1\ \boldsymbol{c}_2\ \ldots\ \boldsymbol{c}_N] \in \mathbb{R}^{N \times N}$ is the matrix whose ith column corresponds to the sparse representation of

sparse subspace clustering

- Clustering Using Sparse Coefficients

To address this problem, we build a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$, where $\mathcal{V}$ denotes the set of N nodes of the graph corresponding to N data points and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges between nodes. $\boldsymbol{W} \in \mathbb{R}^{N \times N}$ is a symmetric nonnegative similarity matrix representing the weights of the edges, i.e., node i is connected to node j by an edge whose weight is equal to w_{ij}. An ideal similarity matrix $\boldsymbol{W}$, hence an ideal similarity graph $\mathcal{G}$, is one in which nodes that correspond to points from the same subspace are connected to each other and there are no edges between nodes that correspond to points in different subspaces.

sparse subspace clustering

Algorithm 1 : Sparse Subspace Clustering (SSC)

Input: A set of points $\{\boldsymbol{y}_i\}_{i=1}^N$ lying in a union of n linear subspaces $\{\mathcal{S}_i\}_{i=1}^n$.

1: Solve the sparse optimization program (5) in the case of uncorrupted data or (13) in the case of corrupted data.
2: Normalize the columns of $\boldsymbol{C}$ as $\boldsymbol{c}_i \leftarrow \frac{\boldsymbol{c}_i}{\|\boldsymbol{c}_i\|_\infty}$.
3: Form a similarity graph with N nodes representing the data points. Set the weights on the edges between the nodes by $\boldsymbol{W} = |\boldsymbol{C}| + |\boldsymbol{C}|^\top$.
4: Apply spectral clustering [26] to the similarity graph.

Output: Segmentation of the data: $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_n$.

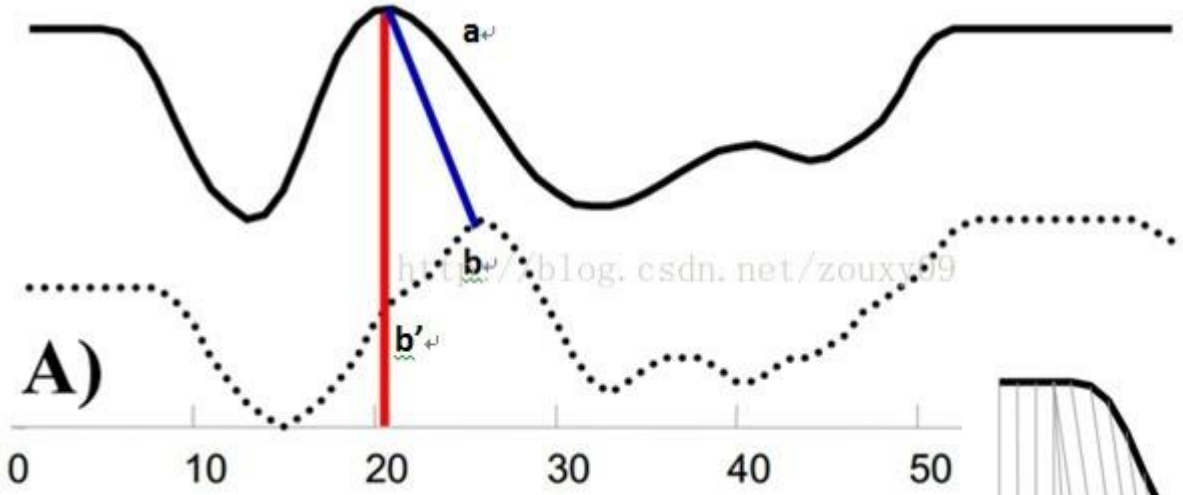
The similarity graph built this way has ideally n connected components corresponding to the n subspaces, i.e.,

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_1 & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{W}_n \end{bmatrix} \boldsymbol{\Gamma}, \tag{6}$$

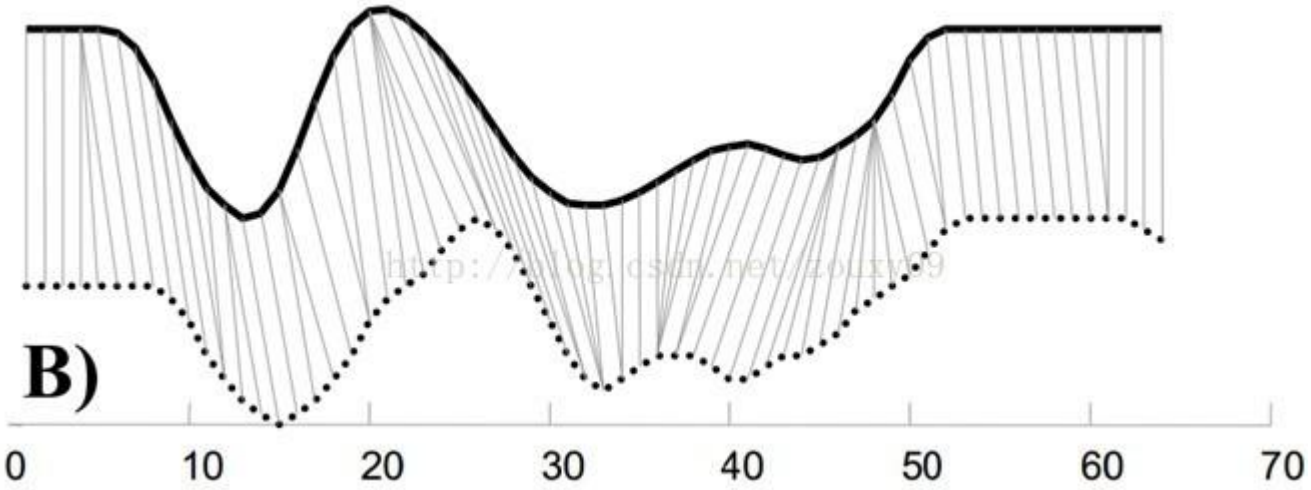
where $\boldsymbol{W}_\ell$ is the similarity matrix of data points in $\mathcal{S}_\ell$.

Functional subspace clustering

- How to construct the adjacency matrix of functional data?



$$\cos\theta = \frac{x_1y_1 + x_2y_2 + \cdots + x_{6400}y_{6400}}{\sqrt{x_1^2 + x_2^2 + \cdots + x_{6400}^2} \cdot \sqrt{y_1^2 + y_2^2 + \cdots + y_{6400}^2}}$$



Functional subspace clustering

manifold. Specifically, for every $X_i \in S_\ell$, we can write the following generalization of the self-expressive equation

$$X_i = \sum_{X_j \in S_\ell, j \neq i} \beta_j d_j(X_j), \tag{3}$$

with some deformation $d_j \in \mathcal{D}$ and scalars $\beta_j \in \mathbb{R}$. A proof is provided in Appendix A. Note that our algorithm

$$S_\ell \triangleq \left\{ X \,\middle|\, X = d\Big(\sum_{\phi_k \in \Phi_\ell} \alpha_k \boldsymbol{\phi}_k \Big); \alpha_k \in \mathbb{R}, d \in \mathcal{D} \right\}, \tag{2}$$

where $\boldsymbol{\phi}_k$ are the basis functions and the set $\mathcal{D}$ contains all possible deformation operators d. We denote the set of all

Given the result in Eq. (3), the cluster assignments of the functional data generated according to Eq. (2) can be uncovered using a novel variant of sparse subspace clustering. We solve the following sparse regression problem for all functions $Y_1, \ldots, Y_n$:

$$\widehat{B}_{i,:} = \underset{B_{i,:}, \{d_j\}}{\operatorname{argmin}} \left\| Y_i - \sum_{j \neq i} B_{i,j} d_j(Y_j) \right\|_2^2, \tag{4}$$
$$\text{subject to} \quad \|B_{i,:}\|_0 \leq s.$$

where $B \in \mathbb{R}^{n,n}$. The L_0 sparsity pseudo-norm indicates the number of non-zero elements of a vector. The goal of this regression is to find the best sparse approximation for Y_i by selecting a few functions Y_j, deforming them by optimizing d_j, and scaling them by multiplying with $B_{i,j}$. After solving Eq. (4) for all functions, similar to subspace clustering we define the symmetric affinity matrix $\boldsymbol{A} = |\boldsymbol{B}| + |\boldsymbol{B}|^\top$ and apply spectral clustering (Ng

Functional subspace clustering

- **Algorithm**

How to determine the value of ϵ.

How to understand the residual R?

Algorithm 1: Functional subspace clustering.

Data: Noisy functional observations $\{Y_i\}_{i=1}^n$ and a termination criteria ϵ.

Result: Clustering assignments for $Y_i, i = 1, \ldots, n$.

for $i = 1, \ldots, n$ **do**
  Initialize $\mathcal{F} \leftarrow \varnothing,\ \mathcal{J} \leftarrow \{i\}\ , R_1 \leftarrow Y_i\ , l \leftarrow 1$
  while $\max_{j \notin \mathcal{J}, d_j} \frac{|\langle R_l, d_j(Y_j)\rangle|}{\|d_j(Y_j)\|_2 \|R_l\|_2} > \epsilon$ **do**
    $\widehat{\boldsymbol{\phi}}_j \leftarrow \operatorname{argmax}_{j \notin \mathcal{J}, d_j} \frac{|\langle R_k, d_j(Y_j)\rangle|}{\|d_j(Y_j)\|_2}$
    $\mathcal{F}_l \leftarrow \mathcal{F}_{l-1} \cup \{\widehat{\boldsymbol{\phi}}_j\}$
    $\mathcal{J} \leftarrow \mathcal{J} \cup \{j\}$
    $\widehat{B}_{i,:} \leftarrow \operatorname{argmin}_{B_{i,:}} \|Y_i - \sum_{\boldsymbol{\phi}_j \in \mathcal{F}_l} B_{i,j} \boldsymbol{\phi}_j\|_2^2$
    $R_{l+1} \leftarrow Y_i - \sum_{\boldsymbol{\phi}_j \in \mathcal{F}_l} \widehat{B}_{i,:} \boldsymbol{\phi}_j$
    $l \leftarrow l + 1$
  end
end
$\boldsymbol{A} \leftarrow |\boldsymbol{B}| + |\boldsymbol{B}|^\top$
Apply spectral clustering to $\boldsymbol{A}$ (e.g., Algorithm 2 in Appendix C) to obtain cluster assignments.

Functional subspace clustering

- deformation operator d

$$d^\star = \operatorname{argmax}_d |\langle Y_1, d(Y_2)\rangle| / \|d(Y_2)\|_2$$

For simplicity, we assume that the length of the time series R_i and Y_j are equal to T_1 and T_2, respectively. In order to efficiently find the optimal warping for a time series Y_i, we note that every warping is an assignment of each point in Y_j to one point in R_i. Thus, we can use a list of binary indicator vectors $Z = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{T_1})$, $\boldsymbol{z}_k \in \{0,1\}^{T_2}$ to represent every deformation as $d(Y_j) = (\boldsymbol{z}_1^\top Y_j, \ldots, \boldsymbol{z}_{T_1}^\top Y_j)$. Now we can reformulate the warping selection process as an integer program

$$\{\boldsymbol{z}_k^\star\} = \operatorname*{argmin}_{\{\boldsymbol{z}_k\}} \frac{\sum_{k=1}^{T_1} (\boldsymbol{z}_k^\top Y_j)^2}{\left(\sum_{k=1}^{T_1} R_{ik} \boldsymbol{z}_k^\top Y_j\right)^2} \tag{5}$$

$$\text{s.t.} \quad z_{k,\ell} \in \{0,1\}, \quad \sum_{\ell}^{T_2} z_{k,\ell} = 1.$$

Spectral clustering

- Spectral clustering is not a simple PCA plus kmeans
- Selection of k

clustering is the eigengap heuristic, which can be used for all three graph Laplacians. Here the goal is to choose the number k such that all eigenvalues $\lambda_1, \ldots, \lambda_k$ are very small, but λ_{k+1} is relatively large. There are several justifications for this procedure. The first one is based on perturbation theory, where we observe that in the ideal case of k completely disconnected clusters, the eigenvalue 0 has multiplicity k, and then there is a gap to the $(k+1)$th eigenvalue $\lambda_{k+1} > 0$. Other explanations can be given by spectral graph theory. Here, many geometric invariants of the graph can be expressed or

Reference

1. Subspace Clustering for High Dimensional Data: A Review
2. MAFIA:efficient and scalable subspace clustering for very large data sets
3. http://blog.csdn.net/WOJIAOSUSU/article/details/58251769?locationNum=11&fps=1
4. Sparse Subspace Clustering: Algorithm, Theory, and Applications
5. https://www.cnblogs.com/Daringoo/p/4095508.html （DTW）
6. Functional Subspace Clustering with Application to Time Series
7. A Tutorial on Spectral Clustering
8. http://blog.codinglabs.org/articles/pca-tutorial.html
9. http://www.taodocs.com/p-478979.html